

Lista de Exercícios 9 - Pacotes Estatísticos

Cristiano de Carvalho Santos

- Poderá ser feito em individualmente ou em dupla.
- Forma de entrega: Submeter no moodle um arquivo “.R” com os comandos utilizados na resolução da lista de exercícios.
- Salvar arquivo com o primeiro nome do(s) estudante(s) que fizeram a solução da lista. Exemplo: Lista9-AlbertoJoana.R.
- Utilize o modelo de resolução disponibilizado.
- Listas enviadas com nome incorreto do arquivo serão penalizadas em 10% da nota. Listas entregues sem a utilização do modelo de resolução serão penalizadas em 20% da nota.
- Prazo de entrega definido no moodle.

Exercícios

Considere o banco de dados sobre a Covid19 para fazer os exercícios.

#=====

Questão 1

- a) Utilizando o ggiraph faça um gráfico de linhas para a evolução dos novos casos de covid19 do estado de Minas Gerais. Adicione os pontos para cada data de forma interativa e que mostrem a quantidade de novos casos.
- b) Faça um gráfico de linhas comparando a evolução de número de casos acumulados em minas gerais e na Bahia.

```
#=====
library(ggplot2)
library(ggiraph)
require(magrittr)
require(readr)

covid_19 <- read_csv("./dados_covid19.csv")

dados1 <- subset(covid_19, state == "MG")
q1a <- dados1 %>% ggplot(aes(date, newCases, color = state)) + geom_line() +
  geom_point_interactive(aes(tooltip=newCases))
ggiraph(code = print(q1a))

dados2 <- subset(covid_19, state == "MG" | state == "BA")
q1b <- dados2 %>% ggplot(aes(date, accumCases, color=state)) + geom_line() +
  geom_point_interactive(aes(tooltip=newCases))
ggiraph(code = print(q1b))
```

```
q1a <- ggplot(codling, aes(x = tot, y = dead, colour = Cultivar, shape = year)) +
  geom_point()
q1a
#=====
```

Questão 2

Utilizando o pacote rgl, faça um gráfico interativo da função

$$f(x, y) = \frac{\cos(x^2)}{4} - \frac{y^2}{9}.$$

```
#=====
require(rgl)

f = function(x, y){
  z = ((cos(x^2)/4) - ((y^2)/9))
}

plot3d(f, col = colorRampPalette(c("blue", "white")),
  xlab = "X", ylab = "Y", zlab = "Z",
  xlim = c(-3, 3), ylim = c(-3, 3),
  aspect = c(1, 1, 0.5))
#=====
#=====
```

Questão 3

Utilizando o pacote gganimate, faça uma animação com um gráfico de barras que ilustre a evolução do número de casos acumulados em todos os estados do Brasil.

```
#=====
library(gganimate)

q3 <- ggplot(covid_19, aes(x=accumCases, y =state, fill = state)) + geom_col()
anim = q3 + transition_time(date) +
  labs(title = "Data: {frame_time}") # Melhor omitir legenda neste gráfico.

anim
#=====
#=====
```

Questão 4

Escolha uma data entre as disponíveis no banco de dados sobre Covid19 e faça um mapa do Brasil de tal forma que cada estado é colorido de acordo com o número de óbitos registrados por covid19 (accumCases) até a data escolhida. Para isto siga os seguintes passos:

1. Pesquise no pacote geobr como obter os dados dos polígonos dos estados do Brasil.
2. Use a função subset ou outra similar para filtrar o banco de dados apenas para data escolhida.

3. Pesquise e use a função “inner_join” para criar um banco de dados com a informação do número total de casos por estado e com os polígonos de cada estado.
4. Adapte o exemplo apresentado no vídeo da aula.

```
#####
require(geobr)
require(leaflet)
library(RColorBrewer)

states <- read_state()
dados4 = subset(covid_19, date == "2020-05-27")
uf_map <- states %>% inner_join(dados4, c("abbrev_state" = "state"))

pal <- colorBin("Blues", domain = log2(uf_map$accumDeaths + 1), bins = 5)

leaflet(data = states) %>%
  addPolygons(fillColor = ~ pal( log2(uf_map$accumDeaths+1) ),
             fillOpacity = 1,
             color = "#BDBDC3",
             layerId = ~ name_state,
             weight = 1,
             popup = paste( uf_map$name_state, "<br>",
                           "Total de óbitos: ", uf_map$accumDeaths)) %>%

  addLegend("bottomright", pal = pal, values = ~log2(uf_map$accumDeaths + 1),
            title = "Escala ", opacity = 1,
            labFormat=labelFormat(transform = function(x)2^(x)-1, digits = 1)) %>%
  addControl("Clique no mapa para ver detalhes", position = "topright")
#####
```

```
#####
```

Questão 5

Utilizando o pacote ggvis e o banco de dados sobre covid19, faça 2 gráficos interativos utilizando 4 tipos diferentes de funções ‘input’. **Obs:** Não tente usar os inputs para escolher a(s) variável(is) utilizados no gráfico pois, aparentemente, isto não está funcionando na versão atual do pacote.

```
#####
require(readr)
require(ggvis)

# solução 1
covid_19 <- read_csv("..\dados_covid19.csv")
covid_19 %>%
  ggvis(~newCases) %>%
  layer_histograms(width = (input_slider(20,200,step=20,label = "Largura")),
                  fill:=input_select(c("Vermelho"="red", "Amarelo"="yellow",

# solução 2
borda <- input_checkbox(label = "Borda branca",
  map = function(val) if(val) "white" else "red")
```

```

cores = input_radiobuttons(
  c("Laranja"="orange", "Verde"="green", "Rosa"="pink", "Azul"="blue"), label="Cores")

covid_19%>% ggvis(x=~newDeaths) %>% layer_histograms(fill:=cores, stroke := borda)

# solução 3
transp = input_checkbox(label = "Pontos semi-transparentes",
  map = function(val) ifelse(val, .3, 1))

combccores = input_checkboxgroup(
  choices = c("Red" = "r", "Green" = "g", "Blue" = "b"),
  label = "Cor do pontos",
  map = function(val) { rgb(0.8 * "r" %in% val, 0.8 * "g" %in% val, 0.8 * "b" %in% val)})

covid_19%>% ggvis(~date, ~newCases)%>%
  layer_points( opacity := transp, fill := combcores)

# solução 4
variaveis = input_radiobuttons(label = "Variável", choices = c("newCases", "newDeaths"), map=as.name)

p <- ggvis(covid_19, y = variaveis)
layer_points(p, x = ~date, fill:=combccores)

#=====
#=====

```

Questão 6

- Utilizando o pacote plotly e o banco de dados iris, faça um gráfico de dispersão entre as variáveis tamanho da Sepala e comprimento da petala, e coloque a espécie no texto interativo
- Utilizando o pacote plotly, faça uma animação (mudando de acordo com a data) de um gráfico de dispersão entre o número de novos casos e novos óbitos de covid para os estados da região nordeste.

```

#=====
library(plotly)
library(tibble)
iris
fig <- plot_ly(iris, x = ~Sepal.Length, y = ~Petal.Length, text = ~Species,
  type = 'scatter', mode = 'markers', marker = list(opacity = 0.5))
fig <- fig %>% layout(title = 'Tamanho da sepala e da petala')
fig

p1 <- subset(covid_19, region == "Nordeste") %>% ggplot( aes(newCases, newDeaths, color = state)) +
  geom_point( aes(frame = as.factor(date), ids = state))
p <- ggplotly(p1)
p
#=====
#=====

```

Questão 7

Utilize a função `facet_trelliscope` do pacote `trelliscopejs` para visualizar os número de novos casos de covid por dia separando curvas por estados em painéis para cada região.

```
#####  
library(trelliscopejs)  
ggplot(covid_19, aes(date, newCases, color = state)) + geom_line() +  
  facet_trelliscope(~ region, nrow = 2 , ncol = 3, width = 600)  
#####
```